[0075] What is claimed is:

1. A method comprising:

enabling renaming of a source of a particular micro-operation even though two or more pointers – each currently indicating where values of a respective group of bits of said source will be found when said particular micro-operation is executed – do not all point to the same register.

2. The method of claim 1, wherein enabling renaming of said source comprises:

generating one or more micro-operations to merge said values into a single register; and

inserting said one or more micro-operations into a sequence of micro-operations that includes said particular micro-operation.

3. The method of claim 2, wherein generating said one or more micro-operations comprises generating two micro-operations when said two or more pointers are three pointers that currently point to three registers, respectively.

4. A method comprising:

generating and inserting into a sequence of micro-operations one or more new micro-operations that merge values distributed among more than one register into a single register.

5. The method of claim 4, wherein generating said one or more new micro-operations comprises:

determining said one or more new micro-operations based at least on pointers to registers where said values will be stored when micro-operations, results of which include said values, are executed.

6. The method of claim 4, wherein generating said one or more new micro-operations comprises generating two micro-operations when said values are distributed among three different registers.

7. A method comprising:

enabling renaming of sources of micro-operations appearing later in a sequence of micro-operations than a particular micro-operation without having to wait for values of bits of a source of said particular micro-operation to be calculated, even though two or more pointers – each currently indicating where values of a respective group of bits of said source will be found when said particular micro-operation is executed – do not all point to the same register.

8. The method of claim 7, wherein enabling renaming of sources of said micro-operations appearing later in said sequence comprises renaming said source of said particular micro-operation.

9. The method of claim 8, wherein renaming said source of said particular micro-operation comprises:

generating one or more new micro-operations to merge said values into a single register; and

inserting said one or more new micro-operations into said sequence.

10. A method comprising:

if two or more pointers — each currently indicating where values of a respective group of bits of said source will be found when a particular micro-operation is executed — do not all point to the same register, enabling execution of micro-operations appearing later than said particular micro-operation in a sequence of micro-operations without having to wait for said values to be calculated, if said micro-operations appearing later in said sequence are not dependent upon said source, not dependent upon said values, and not dependent upon a result of said particular micro-operation.

11. The method of claim 10, wherein enabling execution of said micro-operations appearing later in said sequence comprises renaming said source of said particular micro-operation.

12. The method of claim 11, wherein renaming said source of said particular micro-operation comprises:

generating one or more new micro-operations to merge said values into a single register; and

inserting said one or more new micro-operations into said sequence.

13. A method comprising:

allocating a first register to store results of a first micro-operation a destination of which is an entire architectural register;

subsequently allocating a second register to store results of a second micro-operation a destination of which is a larger partial register of said architectural register; and

when a third register is about to be allocated to store results of a third micro-operation a destination of which is a smaller partial register contained in said larger partial register, generating a merge micro-operation to merge into a single register a) values stored in said first register that correspond to bits of said architectural register but not to bits of said larger partial register and b) values stored in said second register that correspond to bits of said larger partial register but not to bits of said smaller partial register, and inserting said merge micro-operation ahead of said third micro-operation into a sequence of micro-operations including at least said first micro-operation, said second micro-operation and said third micro-operation.

14. The method of claim 13, further comprising:

after inserting said merge micro-operation into said sequence, allocating said third register to store said results of said third micro-operation.

15. The method of claim 14, further comprising:

updating pointers to indicate that values that correspond to bits of said architectural register but not to bits of said smaller partial register will be stored in said single register when said merge micro-operation is executed and values that correspond to bits of said smaller partial register will be stored in said third register when said third micro-operation is executed.

16. A method comprising:

updating a first pointer and a second pointer to point to a first register allocated to store results of a first micro-operation a destination of which is all bits of an architectural register;

subsequently updating said second pointer to point to a second register allocated to store results of a second micro-operation a destination of which is an aligned partial register of said architectural register; and

subsequently renaming a source of a third micro-operation using information represented by said second pointer where said source is a misaligned partial register contained in said aligned partial register.

17. The method of claim 16, further comprising:

subsequently updating said first pointer to point to a third register allocated to store results of a fourth micro-operation a destination of which is said misaligned partial register.

18. The method of claim 17, further comprising:

subsequently renaming a source of a fifth micro-operation using information represented by said first pointer where said source of said fifth micro-operation is said misaligned partial register.

19. A processor comprising:

an architectural register; and

a register tracking mechanism to maintain pointers that indicate where results of micro-operations that are to be written to said architectural register upon retirement will be stored when said micro-operations are executed,

wherein said pointers include a pointer that indicates where results of a most recently allocated micro-operation that writes to all bits of said architectural register upon retirement are to be stored when said micro-operation is executed.

20. The processor of claim 19, further comprising:

a stall detection and micro-operations injection unit a) to identify that said architectural register is a source of a particular micro-operation and that two or more pointers – each to indicate where values of a respective group of bits of said source will be found when said particular micro-operation is executed – do not point to the same register, b) to generate one or more new micro-operations to merge said values into a single register, and c) to insert said one or more new micro-operations into a sequence of micro-operations that includes said particular micro-operation.

21. The processor of claim 19, further comprising:

a stall detection and micro-operations injection unit a) to identify that a partial register of said architectural register is a source of a particular micro-operation and that two or more pointers – each to indicate where values of a respective group of bits of said source will be found when said particular micro-operation is executed – do not point to the same register, b) to generate a new micro-operation to merge said values into a single register, and c) to insert said new micro-operation into a sequence of micro-operations that includes said particular micro-operation.

22. A processor comprising:

a register alias table and allocation unit to eliminate potential renaming stalls of said register alias table and allocation unit due to a sequence of micro-operations that includes a first micro-operation that writes to a partial register and a second micro-operation that reads from a larger register containing said partial register, where said second micro-operation appears later in said sequence than said first micro-operation.

23. The processor of claim 22, wherein said register alias table and allocation unit comprises:

a stall detection and micro-operations injection unit to generate and insert one or more new micro-operations into said sequence.

24. The processor of claim 22, wherein said register alias table and allocation unit comprises:

a register tracking mechanism to maintain pointers that indicate where results of micro-operations, that are to be written to all or part of said larger register upon retirement, will be stored when said micro-operations are executed.

25. An apparatus comprising:

a voltage monitor; and

a processor including at least:

an architectural register; and

a register tracking mechanism to maintain pointers that indicate where results of micro-operations, that are to be written to said architectural register upon retirement, will be stored when said micro-operations are executed,

wherein said pointers include a pointer that indicates where results of a most recently allocated micro-operation that writes to all bits of said architectural register upon retirement will be stored when said micro-operation is executed.

26. The apparatus of claim 25, wherein said processor further comprises:

a stall detection and micro-operations injection unit a) to identify that said architectural register is a source of a particular micro-operation and that two or more pointers – each to indicate where values of a respective group of bits of said source will be found when said particular micro-operation is executed – do not point to the same register, b) to generate one or more new micro-operations to merge said values into a single register, and c) to insert said one or more new micro-operations into a sequence of micro-operations that includes said particular micro-operation.

27. The apparatus of claim 25, wherein said processor further comprises:

a stall detection and micro-operations injection unit a) to identify that a partial register of said architectural register is a source of a particular micro-operation having a source including a lower portion of bits of said architectural register and that two or more pointers – each to indicate where values of a respective group of bits of said source will be found when said particular micro-operation is executed – do not point to the same register, b) to generate a new micro-operation to merge said values into a single register, and c) to insert said new micro-operation into a sequence of micro-operations that includes said particular micro-operation.

28. An apparatus comprising:

    a voltage monitor; and

    a processor including at least:

        a register alias table and allocation unit to eliminate potential renaming stalls of said register alias table and allocation unit due to a sequence of micro-operations that includes a first micro-operation that writes to a partial register and a second micro-operation that reads from a larger register containing said partial register, where said second micro-operation appears later in said sequence than said first micro-operation.

29. The apparatus of claim 28, wherein said register alias table and allocation unit comprises:

    a stall detection and micro-operations injection unit to generate and insert one or more new micro-operations into said sequence.

30. The apparatus of claim 28, wherein said register alias table and allocation unit comprises:

    a register tracking mechanism to maintain pointers that indicate where results of micro-operations, that are to be written to all or part of said larger register upon retirement, will be stored when said micro-operations are executed.

31. An article comprising a storage medium having stored thereon instructions, which when executed by a machine perform a method comprising:

generating and inserting into a sequence of micro-operations one or more new micro-operations that merge values distributed among more than one register into a single register.

32. The article of claim 31, wherein generating said one or more new micro-operations comprises:

determining said one or more new micro-operations based at least on pointers to registers where said values will be stored when micro-operations, results of which include said values, are executed.

33. The article of claim 31, wherein generating said one or more new micro-operations comprises generating two micro-operations when said values are distributed among three different registers.

34. A processor comprising:

an instruction decoder to decode a macroinstruction into one or more micro-operations; and

means for enabling renaming of a source of a particular micro-operation even though two or more pointers – each currently indicating where values of a respective group of bits of said source will be found when said particular micro-operation is executed – do not all point to the same register.

35. The processor of claim 34, wherein said means for enabling comprises:

means for generating one or more micro-operations to merge said values into a single register; and

means for inserting said one or more micro-operations into a sequence of micro-operations that includes said particular micro-operation.

36. The processor of claim 35, wherein said means for generating comprises means for generating two micro-operations when said two or more pointers are three pointers that currently point to three registers, respectively.